Convolutional Neural Network for Intermediate View Enhancement in Multiview Streaming

Convolutional Neural Network for Intermediate View Enhancement in Multiview Streaming

Li YU, *Student Member, IEEE,* Tammam TILLO, *Senior Member, IEEE,* Jimin XIAO, *Member, IEEE,* and Marco GRANGETTO, *Senior Member, IEEE*

***Abstract*—Multiview video streaming continues to gain popularity due to the great viewing experience it offers, as well as its availability which has been enabled by increased network throughput and other recent technical developments. User demand for interactive multiview video streaming that provides seamless view switching upon request is also increasing. However, it is a highly challenging task to stream stable and high quality videos that allow real time scene navigation within the bandwidth constraint. In this paper a convolutional neural network (ConvNet) assisted seamless multiview video streaming system is proposed to tackle the challenge. The proposed method solves the problem from two perspectives: first, a ConvNet assisted multiview representation method is proposed, which provides flexible interactivity without compromising on multiview video compression efficiency. Second, a bit allocation mechanism guided by a navigation model is developed to provide seamless navigation and adapt to network bandwidth fluctuations at the same time. These two blocks work closely to provide an optimized viewing experience to users. They can be integrated into any existing multiview video streaming framework to enhance overall performance. Experimental results demonstrate the effectiveness of the proposed method for seamless multiview streaming.**

***Index Terms*—Multiview video streaming, Multiview navigation, Multiview video representation, Convolutional neural network.**

I. Introduction

Nowadays, many famous movies and TV series are produced in 3D format. The prevalence of 3D videos is owing to its immersive vision, depth perception and interactive involvement. With this trend, a great deal of effort has been made to provide on-demand and live 3DTV services to home, making 3D available outside of the cinema. A recent example of such 3D broadcasting channels was provided during the latest Olympic games.

In light of this, much research effort has recently been devoted to fields related to 3D video. Among them, multiview content representation and coding is a vital topic as part of which, compression efficiency, flexibility and interactivity are considered. There are three main types of 3D video, namely stereoscopic video [1]–[3], multiview video [4] and free viewpoint video [5], [6]. Multiview video coding (MVC) is a popular format, which allows the user to interactively switch viewing angle without the necessity to wear special glasses. The MVC format stems from the multiview extension of the H.264 standard [4]. It exploits the statistical dependencies between spatially neighboring views to increase compression efficiency. The multiview extention of the state-of-the-art high efficiency video coding (HEVC) standard (MV-HEVC) is also available now [7]–[9]. MV-HEVC exploits the spatial redundancy amongst several views based on the conventional block based compensation mechanism. It is backward compatible with any monoscopic decoder by simply extracting the sub-bitstream of the base view.

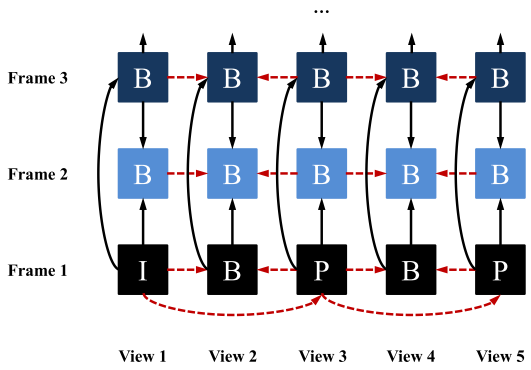

Fig. 1. Illustration of temporal (black arrow) and inter-view (red dashed arrow) prediction in MVC.

The dependencies between the encoding of different views, as shown in Fig. 1, prevents the download of each view separately [10]. For example, if view 4 is required for frame 1, then view $1, 3$ and 5 need to be downloaded. This is because the decoding of view 4 requires view 3 and 5. Moreover, view 3 needs view 1 for decoding. When considering a video with numerous views, most of the downloaded views are left unwatched. This may result in a significant waste of resources in the presence of limited bandwidth and/or computational power. Furthermore, the predetermined bitrates allocated for each view in MVC make it unsuitable for scenarios with heterogeneous bandwidths. As an example, the bitrate of each view cannot be adaptively tuned based on its probability of being watched, as well as the bandwidth level. Simulcast encoding would be an option that allows selective downloading of views, since each view is independently encoded [11]. The stored data volume increases because of unexploited redundancies, however the number of views that need to be

This work was supported by the National Natural Science Foundation of China (NO. 61210006 and NO.61501379) and the Jiangsu Science and Technology Programme (BK20150375). (Corresponding author: Jimin XIAO.)

Li Yu and Jimin Xiao are with the Department of Electrical and Electronic Engineering, Xian Jiaotong-Liverpool University, Suzhou, P.R. China (E-mail:li.yu12@xjtlu.edu.cn; jimin.xiao@xjtlu.edu.cn).

T. Tillo is with the Faculty of Computer Science, Libera Universit di Bolzano-Bozen, Italy.

M. Grangetto is with the Department of Computer Science, University of Torino, Torino 10149, Italy (E-mail: marco.grangetto@unito.it).

transmitted can be reduced. Redundancies between views not only increase the burden on storage but also transmission infrastructure [12]. Thus, a multiview video representation that provides a flexible mechanism for selecting views/bitrates, while maintaining good compression efficiency is an important while non-trivial task.

To solve this problem, the incorporation of a convolutional neural network (ConvNet) module is proposed to enhance video quality in the multiview simulcast encoding framework. The simulcast encoding framework allows the flexible selection of any subset of views to be downloaded, while skipping other non-needed views. Meanwhile, a bandwidth adaptation mechanism will be provided by encoding each view at a variable quality level. With this simulcast framework, arbitrary combinations of views/bitrates are admissible. The ConvNet assisted quality enhancement model helps to reduce the redundancies among the selected combination of views/bitrates by exploiting the similarities between them, and consequently reduces the required bandwidth. This is achieved by selecting a high quality main view to combine with other low quality side views. After receiving views with unequal qualities at the client side, the low quality views are recovered/enhanced thanks to the high quality main view and the ConvNet assisted quality enhancement model. Thus, an equivalent quality is ensured with less bits being transmitted. In summary, this method not only maintains the flexibility of selecting views/bitrates, but also reduces the total number bits transmitted.

Downloading all views that might be watched is crucial for seamless view switching in an interactive multiview system. Therefore, the range of such views, along with their probabilities, needs to be predicted in advance. Similar to [13], a navigation model based on user behavior is proposed in this paper to guide the download of views that are likely to be watched. Based on viewing probability estimation, the proposed bit allocation mechanism strives to minimize the overall distortion as an optimization problem. This bit allocation method is designed for videos encoded with the proposed multiview video representation method and can adapt the overall bitrate to the varying bandwidth.

The main contributions of this paper are summarized as follows:

1) A multiview video representation method is proposed, which provides the flexibility of choosing views/bitrates while maintaining a good compression efficiency thanks to the use of a ConvNet quality enhancement model. To the best of our knowledge, the idea of incorporating ConvNet into the multiview video representation method is novel.
2) A bit allocation method is proposed, based on the probability of watching each view, which works closely with the proposed multiview video representation method to optimize the overall streaming performance.
3) Experimental results demonstrate the effectiveness of the proposed multiview streaming system, with an overall quality improvement of about 0.6 dB with respect to the reference benchmark.

The paper is organized as follows. Related works are introduced in Section II. Then, the proposed method is described in detail in Section III. Experiments and discussions are presented in Section IV to show the effectiveness of the proposed method. Finally, conclusions are provided in Section V.

II. Related works

A. Multiview Video Representation for Seamless View Switching

Multiview video streaming is a challenging task, since one not only needs to interact with user choices but also needs to adapt to network fluctuations. Moreover, the large volume of data represents another challenge here, in comparison to single view video streaming. Thus, a multiview video representation scheme, which provides random accessibility and satisfies the bandwidth and storage constraints at the same time, is highly desirable. There are many classical representation methods for multiview video, including Multiview Video Coding (MVC) [4] and Multiview Video plus Depth (MVD) [14]. However, these solutions are not very effective in an interactive multiview streaming setup since they require the transmission of an entire set of views. Many studies have been carried out into optimizing what becomes a trade-off between storage, bandwidth and interactivity.

In [15], user position is predicted and more bits are allocated to views that are more likely to be watched. Meanwhile, other views are sent in a highly compressed low quality format to aid in possible view switching. This method allows random view switching while reducing the overall bitrate by encoding each view in an unequal fashion. In [16], special switching SP/SI frames [17] are employed to adapt the structure of inter-view predictions for adaptive view switching. In [18], a redundant representation based on I-, P-, and merge frames is proposed with each original picture encoded into multiple versions. The new frame structure enables random access to different views, while maintaining good compression performance. In [19], a Markovian view-switching model with memory is constructed to accurately capture the viewers' behavior. Based on the user behavior model constructed, frame structures are optimized to facilitate periodic view switching so as to achieve an optimal tradeoff between storage cost and expected transmission rate. However, for all of the methods mentioned, several combinations of views/bitrates need to be pre-encoded on the server side, requiring high computational cost and storage requirements. Moreover, the fixed prediction structure still limits the level of adaptivity. [20] anticipates the user's navigation pattern and sends auxiliary information that guarantees temporal and interview consistency. This method shifts the burden due to interactivity from the client side to the server side. Clearly, this solution requires an additional transmission cost. A navigation domain representation for interactive multiview video is proposed in [10], with the aim of providing high flexibility for interactive streaming while maintaining a compression performance similar to that of the classical inter-view predictive coding. It divides the navigation domains into segments that contain a reference frame and some auxiliary information. These navigation segments could render any virtual view within the sub-domain, thus

providing some flexible navigation capacity. [21] proposed an optimization framework for joint view and rate scalable coding of multiview video content represented in the texture plus depth format.

Compared to the aforementioned works, the proposed method i) does not restrict the interactivity and adaptivity of the multiview system, because it does not rely on the interview dependency on the encoder side and ii) it guarantees a flexible and satisfactory quality on the decoding end, leveraging the novel ConvNet based model. At the same time, the bitrate can be flexibly adapted to the bandwidth constraint.

B. ConvNet for Quality Enhancement

ConvNets have been successfully deployed for high level computer vision tasks, such as image classification [22], [23]. Recently, it turns out that the same machinery can be used for pixel level operations, as well. For example, ConvNets have shown better performance than traditional methods for image restoration tasks such as deblocking [24] and restoration [25].

The artifact reduction convolutional neural network (AR-CNN) presented in [26] can effectively attenuate different compression artifacts with a single shallow network, including blocking artifacts, ringing effects and blurring. Inspired by the AR-CNN work, deeper ConvNet modules have been proposed such as the D^3 model [27] and the very deep RED-Net (residual encoder-decoder networks) [28]. The D^3 model is highly effective and efficient owing to the successful combination of both JPEG prior knowledge and sparse coding expertise. The RED-Net adds skip connections between corresponding convolutional and deconvolutional layers and it is capable of handling different levels of noise in a single model. Another work called DnCNN [29], that is designed mainly for Gaussian denoising, can also be used to aid in the removal of image deblocking artifacts. However, like the D^3 model and the RED-Net, it has a very deep architecture and corresponds to a large volume of parameters and high computational complexity. Thus, the 4-layer AR-CNN network is used as the reference network architecture in this paper to maintain a satisfactory performance and a reasonable number of parameters.

The above methods enhance the quality of a single image. While in some cases, there are multiple sources of information that can be exploited for the enhancement, such as consecutive frames within a video [30] and neighboring views for a multiview video [31]. In [30], motion compensation predictions from several frames are used as the input to the ConvNet. In the first stage, two motion compensation algorithms with nine different parameter settings have been utilized to calculate super-resolution drafts. In the second stage, all drafts are combined using a ConvNet. The work in [31], similarly proposes a ConvNet based super-resolution method for multiview videos with mixed resolutions. Low resolution images are up-sampled by taking information from both interplolated low quality images and projected virtual images. Inspired by these two super-resolution works, two inputs are fed into the ConvNet (the projected view with higher quality and a low quality view) to enhance the low quality view.

TABLE I
DESCRIPTIONS OF KEY SYMBOLS

Symbol	Definition
τ	duration of one segment
K	total number of timely non-overlapping segments for one video
k	index of segments whose content is within period $[(k-1)\times\tau, k\times\tau), 1\le k\le K$
M	total number of available quality levels
N	total number of available views
t_k	decision point of k^{th} segment
v_i	the i^{th} view among all the available views, $1\le i\le N$
q_j	the j^{th} quality level among all available representations, $1\le j\le M$ (larger j means better quality level)
$B_k^{'}$	the predicted bandwidth when downloading ϕ_k
$V(t_k)$	the view angle of user at decision point t_k
$R_k(v_i,q_j)$	the bitrate of texture segment $S_k(v_i,q_j)$
$R_k^d(v_i)$	the bitrate of depth segment $D_k(v_i)$
$d_k(v_i)$	the k^{th} depth map of view v_i
$S_k(v_i,q_j)$	the k^{th} texture segment of view v_i and quality level q_j
ϕ_k	set of k^{th} segments requested by client, including n texture segments and one depth map of the central view (v_c): $\phi_k=\{S_k(v_i,q_j)\}_n\cup D_k(v_c)$
$\Re_k$	the estimated available bits for downloading textures in ϕ_k, i.e. $\{S_k(v_i,q_j)\}_n$

C. Dynamic Adaptive Video Streaming over HTTP

Nowadays, MPEG-DASH [32] is widely used for video streaming. A typical DASH system consists of an HTTP sever and a DASH client. They communicate with each other through the HTTP network. DASH transfers the management of streaming from the server side to the client side, which saves a significant amount of the servers resources and permits dynamic adaptivity.

In the HTTP server, video contents of different views, representations and their description files (Media Presentation Description, MPD) are stored. Each video representation is divided in the time domain into several chunks, which are named segments. Each segment is usually 2 seconds long [33]–[35]. Each segment is stored as an independent file, which is associated with a URL address.

The DASH client first obtains the MPD file from the server, and then decides which segment to pull based on network conditions. Generally speaking, the decision mechanism [34]–[38] aims to maximize the average video quality, reducing the frequency of quality switching, as well as avoiding playback freezing [33].

For the streaming method proposed in this paper, bitrate adaptation and quality enhancement are implemented at the decoder side. Therefore, it maintains compatibility with a standard DASH server. This represents another advantage of the method presented here.

III. PROPOSED METHOD

In this paper, a multiview video streaming system that allows real-time and seamless view switching is proposed. The system mainly consists of three parts: the ConvNet assisted quality enhancement model, the navigation model and the bit

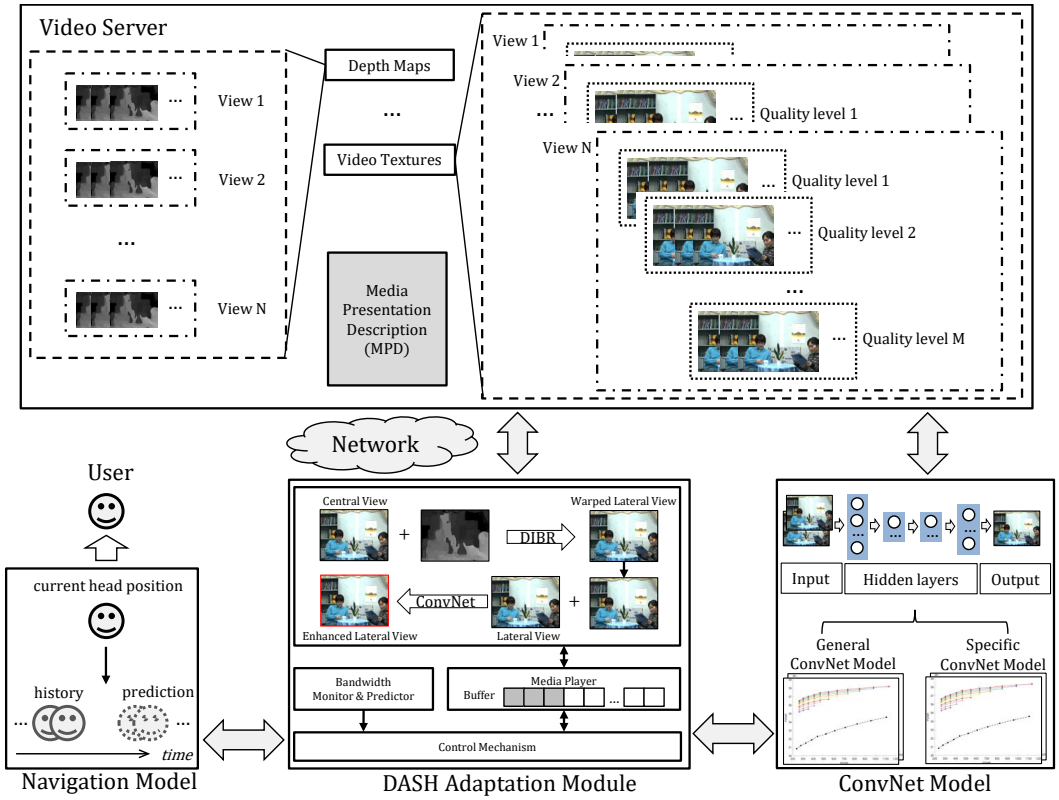

Fig. 2. Block diagram of the proposed multiview video streaming solution (the DASH framework is used in this illustration as an example).

allocation mechanism. Thanks to the ConvNet assisted quality enhancement model, a novel multiview video representation based on simulcast encoding is proposed to jointly ensure viewing flexibility and compression efficiency. Furthermore, the bit allocation mechanism tunes the quality of each selected view according to the navigation model, so as to reduce the overall distortion. The three parts can be incorporated into any multiview streaming framework to provide an enhanced performance.

In the following, the view most likely to be watched is denoted **central view**, while the remaining views are called **lateral views**. It follows the rule that the central view is requested with high quality, while the lateral views are downloaded at lower quality levels.

DASH is used as the streaming framework in the following discussions. First, an overview of the proposed method will be presented, along with definitions of the notations used. Then, detailed descriptions of the ConvNet assisted quality enhancement model, the navigation model and the bit allocation mechanism will be provided, respectively.

A. Solution Overview

The block diagram of the proposed method is illustrated in Fig. 2, which consists of an HTTP server, the navigation model, the DASH adaptation module and the ConvNet model. The last three parts all belongs to the DASH client. The proposed bit allocation mechanism is incorporated into the DASH adaptation module.

In the following, the data preparation on the HTTP sever, the details of the DASH adaptation module and a brief work flow will be described. Important notations and corresponding definitions are listed in Table I.

1) HTTP Server: Suppose there are N views $\{v_1, v_2, \ldots, v_N\}$ provided for one video. The video of a certain view is divided into K segments with a duration of τ s. For each segment, M alternative bitrate/quality representations $\{q_1, q_2, \ldots, q_M\}$ are available. Thus, the k^{th} segment of the view v_i with the quality representation q_j is denoted as $\boldsymbol{S_k(v_i, q_j)}$. It follows that there are $K \times N \times M$ texture video segments for one content, which are independently encoded into separate files.

Besides the texture, the depth maps are also prepared as one adaptation set. As opposed to the texture case, only one representation guaranteeing stable warping performance is prepared. Thus, the k^{th} depth segment of view v_i is denoted as $\boldsymbol{d_k(v_i)}$.

2) DASH Adaptation Module: This module aims to provide the best viewing experience, i.e., video QoE [39]–[41], under bandwidth constraints. To achieve this goal, it has multiple duties. First, it is responsible for coordinating the navigation model and the ConvNet model, as well as the external communication with the server. Second, it is responsible for displaying the video, which includes parsing received data from the server, choosing the view to display according to the user's current viewing angle and enhancing the lateral view with the ConvNet model if the user's viewing angle is switched. Third, the DASH client is responsible for monitoring

and predicting the bandwidth status, as well as the buffer status. Last but not the least, it is responsible for the adaptation intelligence, which decides which data to request, based on the available bandwidth and the user's navigation mode.

3) Workflow in Brief: The DASH client operations are described as follows. After obtaining the MPD file from the video server, the download of multiview video starts. Suppose that at the decision point t_k, the client is going to request the desired segment set ϕ_k $(1 \leq k \leq K)$, which contains n views that might be watched and one depth map of the current central view v_c, which can be represented as

$$\phi_k = \{S_k(v_i, q_j)\}_n \cup d_k(v_c). \tag{1}$$

The detailed workflow is as follows:

a) The current head position of the user $V(t_k)$ is tracked by the navigation model. Correspondingly, the current central view will be chosen: $v_c = V(t_k)$.
b) The n views that are possible candidates to be watched, $\{v_i\}_n$, are predicted along with their probabilities, by the navigation model described in Section III-C.
c) The future bandwidth $B_k^{'}$ for downloading ϕ_k is predicted in the DASH adaptation module according to the monitored history. Then, the estimated available number of bits $\Re_k$ is :

$$\Re_k = B_k^{'} \times \tau. \tag{2}$$

d) After subtracting the bits of the depth map from $\Re_k$, the remaining bits are allocated among the n views according to the proposed bit allocation mechanism in Section III-D.
e) The bit allocation proposal, which contains the representation levels for each desired view, would be encapsuled into the request and sent to the server.
f) When the user switches from the central view to a lateral view, the low quality lateral view is enhanced by the ConvNet model described in Section III-B before being displayed.

This procedure is iterated for each segment.

B. ConvNet Model

The proposed ConvNet model aims to improve the overall quality without increasing the total bitrate. This is achieved by enhancing the qualities of the lateral views based on their similarities with the central view.

Inspired by the AR-CNN work [26] and our previous work [31], the architecture of the ConvNet network used in this paper is shown in the Fig. 3. The inputs include the low quality lateral view and the virtual image warped from the high quality central view. The output is the lateral view with enhanced quality. It is worth noting that the ConvNet model is trained with a set of multiview video sequences: our findings shows that the model can also be used to enhance sequences outside of the training set effectively. In the following, the pre-processing of input data, the ConvNet network structure and the training details are introduced.

1) Pre-processing: prepares the data to be fed into the ConvNet, which consists of two steps: one is 3D warping and the other is batch cropping. For 3D warping, the high quality central view and its depth map are used to construct a 3D image, and then generate the virtual view in the position of the requested lateral view. The DIBR technique [42] is used to warp the pixels. In this process, holes might appear due to the fact that some occluded regions become visible. Such holes are usually recovered with inpainting methods; in this case the inpainting step is skipped and the ConvNet learns how to recover the missing areas, as well. Next, input images are cropped into small sub-images (i.e. 33×33), including the low quality lateral image (Y_L), DIBR warped image (Y_V) and the ground truth images. This process helps to speed up the training process. Sub-images at the same position are gathered into a group, then a mapping between the two input images and the ground truth are learned using the ConvNet.

2) ConvNet for quality enhancement: consists of 4 layers. Specifically, the first layer performs image fusion and feature extraction, which fuses the two inputs, i.e. Y_L and Y_V, and extracts feature vectors from them. The second layer is responsible for feature enhancement, which would remove noise and compression artifacts from the feature vectors. Then, the non-linear mapping layer maps the low quality patches into high quality patches. Finally, the reconstruction layer merges the obtained patches to deliver the final output. The network can be represented as:

$$F_1(Y) = max(0, W_{11} * Y_L + W_{12} * Y_V + B_1); \tag{3}$$
$$F_i(Y) = max(0, W_i * F_{i-1}(Y) + B_i), i = 2, 3; \tag{4}$$
$$F(Y) = W_4 * F_3(Y) + B_4. \tag{5}$$

The first and last layers are computed by Eq. (3) and Eq. (5) respectively, while the second and the third layers are determined according to Eq. (4). W_i and B_i denote the filters and biases of the i^{th} layer respectively, and '*' denotes the convolutional operation. The Rectified Linear Unit (ReLU) is applied to the filter responses. F represents the output feature maps. The kernel sizes are 9×9, 7×7, 1×1 and 5×5 for each layer respectively. Further details can be found in the AR-CNN paper [26].

3) Training: ConvNet network training is conducted to learn the filters and biases. Given a training dataset $\{Y_L(i), Y_V(i), Y_G(i)\}$, the Mean Squared Error (MSE) is used as the loss function. Y_L and Y_V represents the input data: the low quality image and the virtual image warped from the high quality view. While Y_G denotes the ground truth image. The loss function can be expressed as:

$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left\| f(Y_L(i), , Y_V(i), \theta) - Y_G(i) \right\|^2, \tag{6}$$

where θ is the parameter set of the network, including both the convolution filter weights and biases; n is the number of training samples. Stochastic gradient decent (SGD) is used to minimize the loss function with the standard backpropagation mechanism. At the beginning, the weights of the network are initialized with values from a random gaussian distribution

$$P_{k+l}(v_{i+j}) = \begin{cases} P_{k+l-1}(v_i) \times p + g(1) \times P_{k+l-1}(v_{i-1}) \times (1-p) & , \quad j = 0; \\ g(0) \times P_{k+l-1}(v_{i+j-1}) \times \frac{(1-p)}{2} + g(1) \times P_{k+l-1}(v_{i+j}) \times p + g(2) \times P_{k+l-1}(v_{i+j+1}) \times \frac{(1-p)}{2}, & \quad j \in (0, l]; \\ P_{k+l}(v_{i-j}) & , \quad j \in [-l, 0). \end{cases} \tag{10}$$

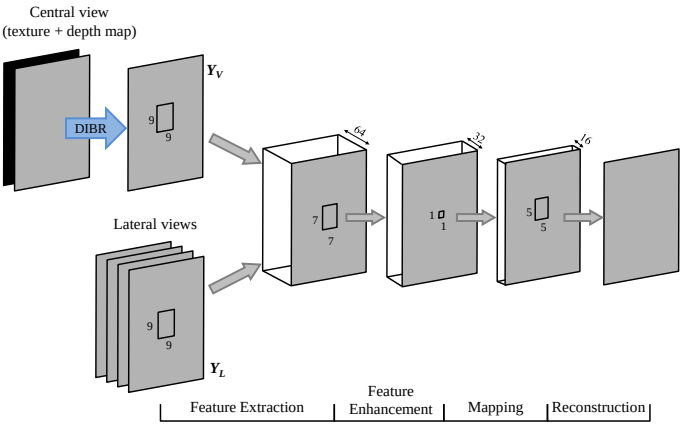

Fig. 3. ConvNet network structure with 4 convolutional layers.

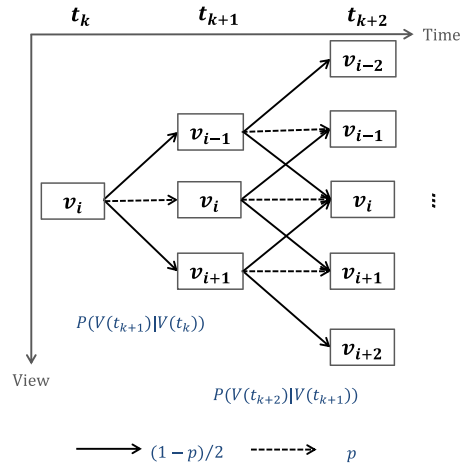

Fig. 4. Graphical representation of user navigation model, where different transition probabilities are represented by different arrow types.

with a mean of 0 and a variance of 0.001. The learning rate is set at 0.0001.

C. Navigation Model

Tracking and predicting the user's head position are vital to a multiview video system, which enables the interactivity between the user and the system. This function can be achieved by a navigation model. This model tracks the motion of the user's head and predicts future possible viewing angles. The prediction information helps the system to prefetch all the views that might be watched in the future, guaranteeing real-time view switching. Let us suppose that at the decision point t_k, the user's view angle is $V(t_k)$. The navigation model employed in this paper is based on the following assumptions:

1) One state assumption: The current view angle $V(t_k)$ is only affected by the previous view angle $V(t_{k-1})$.

2) Smooth view switch assumption: A large view switch step is not allowed. That is, the user can at most switch to the neighboring view in one step, i.e.

$$V(t_k) \in \{v_{i-1}, v_i, v_{i+1}\}, \text{ if } V(t_{k-1}) = v_i. \tag{7}$$

Based on these two assumptions, the navigation model can be illustrated as in Fig. 4. Each block in the figure denotes one possible view angle at that decision point. For example, the view angle at t_k is v_i. At the next decision point t_{k+1}, there are three possible view angles $\{v_{i-1}, v_i, v_{i+1}\}$ to be switched to.

The probability of remaining in the previous view angle is p, while those of switching to the neighboring view angles are both $(1-p)/2$.

$$P(v_i|v_i) \quad = p; \tag{8}$$

$$P(v_i|v_{i-1}) = P(v_i|v_{i+1}) = \frac{1-p}{2}. \tag{9}$$

Let us now consider the prefetching of future segments, such as those up to time t_{k+l}, when the k^{th} segment at view angle $V(t_k)$ is being displayed. Depending on the different values of l, the admissible views for t_{k+l} vary, together with the corresponding probabilities. The probability for each view is the sum of the probabilities of all the possible view switching paths leading to that view. Meanwhile, the probability of each possible view switching path is calculated as the multiplication of the switching probabilities along the path. Thus, once the probability of each step is known, the final probability is easy to calculated.

The probability of watching a viewing angle $V(t_k) = v_i$ at time t_k is denote as $P_k(v_i)$. Then, the probability of watching view v_{i+j} at $t_{k+l}, (l \geq 1)$ is calculated recursively as $P_{k+l}(v_{i+j})$ in Eq. (10), where $g(x)$ is calculated as follows:

$$g(x) = \max(0, \min(1, l - j - x + 1)), x \geq 0. \tag{11}$$

Computation of $P_{k+l}(v_{i+j})$ in Eq. (10) is divided into three cases: $j = 0$ when the view angle remains the same as $V(t_k)$; $j \in (0, l]$ when the view angles are on the right of $V(t_k)$; $j \in [-l, 0)$ when the view angles are on the left of $V(t_k)$. The probability for $j \in [-l, 0)$ is identical to that for $j \in (0, l]$. It can be observed in Fig. 4 that, based on the different values of l and j, the number of switching paths leading to the current view angle varies. The function $g(x)$ is employed to deduce their number based on l and j. In summary, the probability is calculated recursively by Eq. (10).

At the end, the possible views to be watched, along with their probabilities, would be sent to the DASH adaptation module for the further operations.

D. Bit Allocation Mechanism

As one part of the DASH client, the bit allocation mechanism aims to minimize the expected distortion of all views, including the central view and the lateral views that might be watched:

$$\begin{aligned}\bar{D}_k &= \sum_{1\le i\le n} D_k(v_i,q_j)\times P_k(v_i),\\ \text{s.t.} &\sum_{1\le i\le n} R_k(v_i,q_j)\le \Re_k.\end{aligned} \tag{12}$$

The constrained optimization problem can be converted into an unconstrained one using the Lagrange multiplier method:

$$J=\bar{D}_k+\lambda\sum_{1\le i\le n} R_k(v_i,q_j); \tag{13}$$

where λ is the Lagrange multiplier. To minimize J, the derivative of J with respect to the bitrate of each view is set to zero. Taking a given view $v_i(1\le i\le n)$ as an example,

$$\frac{\partial J}{\partial R_k(v_i,q_j)}=P_k(v_i)\times\frac{\partial D_k(v_i,q_j)}{\partial R_k(v_i,q_j)}+\lambda=0. \tag{14}$$

Thus, for any two views, such as v_i and $v_{i'}$,

$$P_k(v_i)\times\frac{\partial D_k(v_i,q_j)}{\partial R_k(v_i,q_j)}=P_k(v_{i'})\times\frac{\partial D_k(v_{i'},q_{j'})}{\partial R_k(v_{i'},q_{j'})}.$$

That is,

$$\frac{P_k(v_{i'})}{P_k(v_i)}=\frac{\partial D_k(v_i,q_j)}{\partial R_k(v_i,q_j)}\Big/\frac{\partial D_k(v_{i'},q_{j'})}{\partial R_k(v_{i'},q_{j'})}. \tag{15}$$

1) Initial Bit Allocation: This step directly works on the outputs of the HEVC codec, without considering the ConvNet enhancement model. Each quality level q_j maps to one QP value without overlapping. As defined in HEVC [43],

$$\lambda_{HEVC}(q_j)=\frac{\partial D}{\partial R}=c*2^{\frac{QP_j-12}{3}}, \tag{16}$$

where c is a parameter related to the coding structure. Since the two frames are in the same position, i.e. the i^{th} frame in a coding structure, the values of c are the same. Thus, from Eq. (15) it follows that:

$$\begin{aligned}P_k(v_{i'})/P_k(v_i) &=\lambda_{HEVC}(q_j)/\lambda_{HEVC}(q_{j'})\\ &=2^{\frac{QP_j-12}{3}}/2^{\frac{QP_{j'}-12}{3}}\\ &=2^{\frac{QP_j-QP_{j'}}{3}},\end{aligned} \tag{17}$$

where each q_j maps to a unique QP_j, i.e. $q_j=F(QP_j)$. Finally, the optimal QP difference between any two views, v_i and $v_{i'}$, turns out to be:

$$\Delta QP(v_i,v_{i'})=QP_j-QP_{j'}=3log_2\frac{P_k(v_{i'})}{P_k(v_i)}. \tag{18}$$

This method can also be used for other codecs, provided that Eq. (16) is substituted accordingly. In conclusion, the QP value of any lateral view can be set as function of the QP of the central view (QP_c):

$$QP_j=QP_c+\Delta QP(v_i,v_c).$$

Then, the bit allocation problem is converted into the problem of finding the minimum value of (QP_c) under the following total bitrate constraint:

$$\sum_{1\le i\le n} R_k(v_i,F(QP_c+\Delta QP(v_i,v_c)))+R_k^d(v_c)\le\Re_k, \tag{19}$$

which can be solved by a simple iterative search method. Till now, the optimal bit allocation has been derived without taking the ConvNet enhancement into account.

Algorithm 1 Fine tuning of bit allocation proposal.

Input:
Total available bits $\Re$;
The initial bit allocation proposal, $QP_c,\{QP_l\}$;
The range of available representations, $[QP_{min},QP_{max}]$;
RD curve of HEVC encoded sequence, RD_o;
RD curve of ConvNet enhanced sequence, RD_e;
Output:
The fine tuned bit allocation proposal, $QP_c^e,\{QP_l^e\}$;

```
Initialization: QP_c^e = QP_c; {QP_l^e} = {QP_l}; ΔR^e = 0;
repeat
    QP_c^tmp = QP_c^e − 1;
    Calculate ∂D_c/∂R_c of QP_c^tmp in RD_o;
    for all {QP_l} do
        Calculate ∂D_l/∂R_l with respect to ∂D_c/∂R_c according to (15);
        Search for the QP_l^tmp within [QP_min, QP_max] in RD_e, which guarantee the nearest slope to ∂D_l/∂R_l;
    end for
    ΔR = Σ R(v_i, F(QP_i^tmp)) + R^d(v_c) − ℜ;
    if ΔR < ΔR^e then
        Update QP_c^e, {QP_l^e} with QP_c^tmp, {QP_l^tmp};
        ΔR^e = ΔR;
    end if
until QP_c^tmp == QP_min
if ΔR^e ≤ 0 then
    Allocate ΔR^e among lateral views according to the quality gain per bit, and update {QP_l^e} accordingly;
end if
return QP_c^e, {QP_l^e};
```

2) Fine Tuning based on the ConvNet Enhancement Model: This step aims at incorporating the influence of the ConvNet quality enhancement into the bit allocation process. The ConvNet clearly modifies the rate distortion (RD) relationship for the quality-enhanced lateral views. However, the tuning is not straightforward since it still needs to try to satisfy Eq. (15). Following the ConvNet quality enhancement, Eq. (16) is no longer valid for the lateral views. Instead, the slope of the tangent line in the modified RD curve is employed. This is based on the assumption that the RD curve is regarded as linear within a local range. Each enhanced lateral view has its own RD curve, which consists of RD values for different QPs. One can estimate the i-th RD slope as:

$$\frac{\partial D_i^e}{\partial R_i}=\frac{D_{i+1}^e-D_{i-1}^e}{R_{i+1}-R_{i-1}}, \tag{20}$$

where D^e_{i+1} and D^e_{i-1} are the distortions of the ConvNet enhanced images when encoded with QP_{i+1} and QP_{i-1}, respectively.

The detailed fine tuning algorithm is shown in Algorithm 1. It starts from the initial bit allocation result and decreases the QP of the central view (QP_c) step by step, until reaching the available minimum QP (QP_{min}). For each QP_c, the QPs for the lateral views ($\{QP_l\}$) are selected based on Eq. (20) and Eq. (15). The $\{QP_l\}$ that best matches Eq. (15) is chosen. Then, the delta bits (ΔR) are calculated between the actually used bits and the total available bits ($\Re$) to see whether it fulfills Eq. (19). If $\Delta R < 0$, this means the actual used bits is within the available bit budget. Thus, this fine tuned proposal is regarded as feasible. This process is iterated for all the possible QP_c, and the one among all feasible proposals that best fits Eq. (19) is selected.

The presented optimization algorithm is not able to precisely satisfy the bit rate budget $\Re$, because of the limited set of rate points determined by different QP values. In considering whether it is better to allocate the extra bits ΔR either to the central or lateral views; the lateral views are encoded in low quality, hence their RD curve slope is usually larger, i.e. a small increase in bit rate can result in a significant distortion reduction: As a consequence, steps $15-17$ in the Algorithm 1 are designed to allocate ΔR among the lateral views according to their quality gain per bit.

IV. Experiments

Multiview video sequences, including Kendo, Balloons, Undodancer and GT-fly [44], are used in the following experiments. Each view is independently encoded with HM 16.0 [45] with different QPs (ranging from 22 to 51) to represent different quality levels. Depth maps are encoded with $QP = 50$.

In the following, the performance of the proposed method is demonstrated using the DASH protocol. It is assumed that the bitrate allocated to each video segment is controlled using a standard DASH implementation. Thus, the following experiments are evaluated at the segment level without taking into account the bandwidth fluctuations from segment to segment. The duration of each segment equals 2 seconds. The proposed method can be applied to any practical DASH system to enhance the performance of each segment, no matter which global rate adaptation mechanism is used. The experiments are arranged as follows: first, the effectiveness of the ConvNet based quality enhancement model is examined to demonstrate the feasibility of the proposed multiview representation method. Next, the proposed bit allocation method is assessed for each segment set in comparison with the benchmark. The benchmark here represents the initial bit allocation mechanism as described in Section III-D. The above experiments are applied on the luminance channel (Y channel in YUV color space), and the performance is evaluated in terms of PSNR.

A. ConvNet Assisted Quality Enhancement Model

The goal is to enhance the quality of the lateral view Y_L through the ConvNet model, with Y_L and the virtual view Y_V as inputs. Y_V is warped from the central view Y_C. The enhancement task is non-trivial, since there are a large number of different scenarios. The performance is affected by many factors, including the compression quality of Y_L, Y_C, the quality difference between them, the direction and the distance (i.e. the baseline) from Y_C to Y_L, as well as the contents of the video. The interest here is in investigating whether the proposed ConvNet model is robust to all of the mentioned aspects.

According to the experiment, the performance of the ConvNet model is influenced by the training set. With a more dynamic data set, which incorporates more scenarios, the corresponding ConvNet model would guarantee a more generalized enhancement result over the different scenarios. While with a more specific data set, the ConvNet model would provide a more effective performance on that specific data. The former is named **general ConvNet model**, while the latter one is called **sequence-specific ConvNet model**. The general ConvNet model is trained with multiple sequences exhibiting different contents and baselines. While the sequence-specific ConvNet model is trained with only one specific sequence.

1) General ConvNet Model: A general ConvNet model is trained with multiple video sequences, as well as different QP values and different distances between lateral and central views. First 15 frames of two sequences, i.e. Kendo and Undodancer, are used as the training data. Two distances are included, namely single and double baselines. The QPs of Y_L are 30 and 42, while the QP of Y_C is 20. The QP of the depth map is fixed at 50. Through experimentation it has been observed that the QP of depth map does not significantly influence the enhancement result (less than 0.1 dB) [46]. The model is trained over 1.5 million iterations.

The performance of the general ConvNet model on different sequences is shown in Fig. 5. Frames from 171 to 200 of each sequence are used as the test data. Different curves are plotted with different QP for Y_C, ranging from 22 to 30. While for the QP of Y_L, its range is deduced from the proposed bit allocation mechanism. Each dot in the figure represents the average PSNR computed on the 30 frames. PSNR of Y_L without enhancement is plotted as the benchmark for comparison. The observations are as follows:

a) The enhancement gain provided by the general ConvNet model over the benchmark is huge, with roughly 5 dB increase at most. It can also be noted that the quality of the enhanced view increases, when increasing the quality of the central view.
b) The general model works well for both camera recorded videos and computer generated videos.
c) The general model is effective for scenarios with different distances and directions between the views. Three scenarios are tested for different sequences in each row in Fig. 5. This demonstrates the stability of the general ConvNet model.
d) Although the video contents of Balloons and GTfly are completely different from the contents of the training data, the results are still promising. This demonstrates the generality of the general ConvNet model.

2) Sequence-specific ConvNet Model: The sequence-specific ConvNet model is trained with a specific sequence to

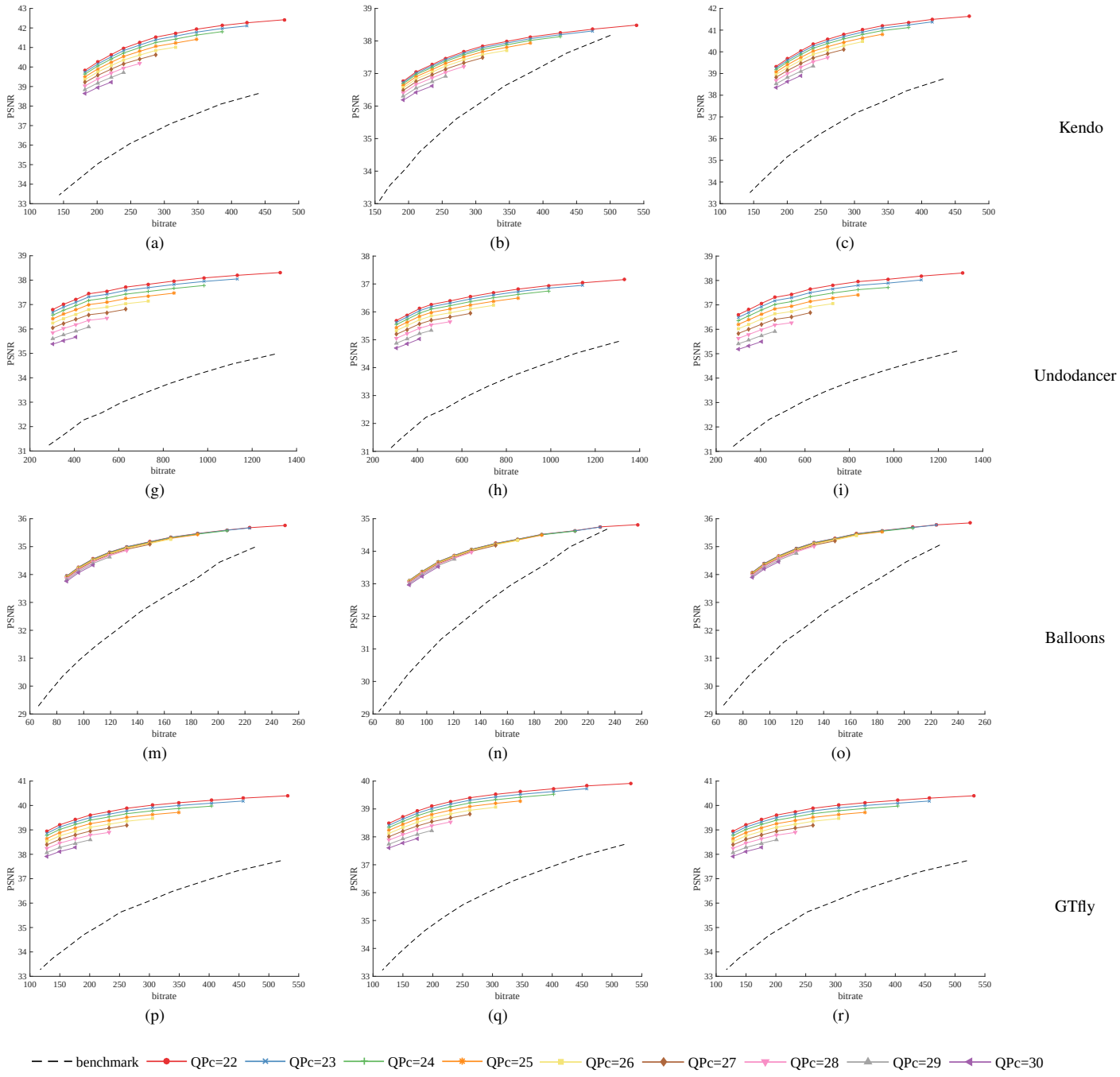

Fig. 5. Rate Distortion curves of **General ConvNet Model**, in comparison with benchmark. The benchmark represents the HEVC encoded sequence without any enhancement. The views on the left column are $Y_L = 2, Y_C = 3$; the center column are $Y_L = 1, Y_C = 3$; the right column are $Y_L = 4, Y_C = 3$. The above two rows are results of sequences within the training set, while the bottom two rows are those of sequences outside the training set.

maximize the gain for that specific data. In Fig. 6, sequence-specific ConvNet models for Kendo and Undodancer are compared to the general ConvNet model respectively. As expected, the sequence-specific ConvNet models yield better results than the general ConvNet model. It is worth noting that the gain for Undodancer is more uniform than that for Kendo across different bitrate levels. For Kendo, the gain of the sequence-specific ConvNet model is obvious in the low bitrate region. This might be owing to the fact that, when the depth data is inaccurate (for the case of the camera recorded sequences whose depth maps are estimated), the sequence-specific model will learn how to mitigate this depth inaccuracy. Whereas for the general ConvNet model, it has also been trained with computer generated sequences whose depth maps are accurate. Thus, it has a lesser ability to compensate the inaccuracy of the depth data. When it comes to the high bitrate region, similar performance is obtained for Kendo with both models. This is because the quality of the warped view suffers from the limited quality of the estimated depth data; consequently, the warped view offers limited contribution to the enhancement of the already good lateral view.

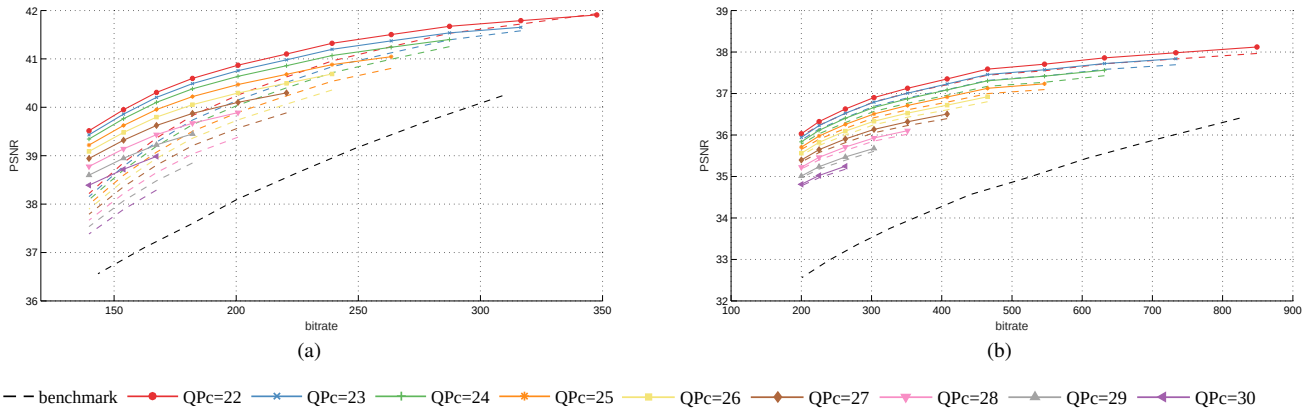

Fig. 6. Rate Distortion curves of **Sequence-specific ConvNet Model** (solid lines), in comparison to General ConvNet Model (dashed lines). The views are $Y_L = 2, Y_C = 3$; The sequences are (a) Kendo; (b) Undodancer.

3) ConvNet model transmission cost: As already pointed out, the parameters of the ConvNet model need to be transmitted to the client at the beginning of the streaming session to ensure the same performance would be obtained at both the encoder and the decoder side. Since the ConvNet model used in this paper is a simple 4-layer network, the number of parameters is limited. In particular, the size of the ConvNet model proposed is 447 kB. Once the ConvNet model is received, it is used to enhance the quality of the whole sequence. It is worth noting that the one-time transmission of the ConvNet parameters is equivalent to less than 0.5% of a ten-minute YouTube video streamed at a resolution of 1280×720. In summary, the transmitted bits of the ConvNet parameters is almost negligible in comparison to the rate-distortion gain obtained with proposed model. Moreover, in the case of the general ConvNet model, it can be used to enhance multiple sequences thus making its overhead negligible.

B. Navigation Guided Bit Allocation Mechanism

In this section, the overall PSNR of the views requested by the user is shown to demonstrate the effectiveness of the proposed bit allocation mechanism. The obtained gain relies on both of the following proposed strategies: i) quality enhancement of the ConvNet model; ii) exploitation of the remaining available bits ΔR. These two stages correspond to Steps $1-14$ and Steps $15-17$ in Algorithm 1.

1) Experiment Setup: Two navigation scenarios are tested, with 3 and 5 views as the view switching range respectively. That is, $\{v_{i-1}, v_i, v_{i+1}\}$ and $\{v_{i-2}, v_{i-1}, v_i, v_{i+1}, v_{i+2}\}$ as shown in Fig. 4. The probability of remaining in the previous view angle (p) is set to 0.9. For Kendo and Undodancer, both general and sequence-specific ConvNet models are used for testing. While for Balloons and GTfly, only the general ConvNet model is assessed.

Supposing view $1-n$ is requested, the overall PSNR is defined as the following weighted average:

$$\overline{PSNR} = \sum_{1 \le i \le n} PSNR(v_i, q_j) \times P(v_i),$$

where $P(v_i)$ is obtained according to Eq. (10). $PSNR(v_i, q_j)$ represents the PSNR of the i^{th} view. The $\overline{PSNR}$ *gain* is calculated as the difference between the $\overline{PSNR}$ of the proposed method and the benchmark.

2) Gains from the ConvNet Assisted Quality Enhancement Model (Steps $1-14$): In order to show the effectiveness of incorporating the ConvNet model, the total number of bits determined by the bit allocation mechanism is constrained to be lower than those of the benchmark. Results for the 3- and 5-view switching ranges are shown in Table. II and Table. III respectively. The benchmark method uses only the initial bit allocation. Based on the total bits used by the benchmark, the proposed method allocates the same bits following steps $1-14$ in Algorithm 1. It is worth noting that the assigned QP values for the various views, by the proposed approach, differ from sequence to sequence. As expected, the QP_c in the proposed method is usually lower than that in the benchmark, while the QP_l has an opposite trend. The overall PSNR's for both methods are listed, with those using the sequence-specific ConvNet model in bold. The overall PSNR gain of the proposed method over the benchmark is listed in the last column. Besides, the average total bits, the average overall PSNR and the average PSNR gain are shown in the bottom of each table, with results of the general ConvNet model and the sequence-specific ConvNet model listed separately. The observations are as follows:

a) For each sequence, three bit levels are tested and compared. The proposed method stably outperforms the benchmark. The general ConvNet model yields 0.6 dB gain on average for both view switching ranges. The gains with the sequence-specific ConvNet model are even higher.
b) The sequence-specific ConvNet model (results in bold) shows its strength in the scenario with 5 views, where a larger gain of about 0.3 dB is documented with respect to the general ConvNet model. It is expected that the gain would become even more significant with more views. This is because the bitrate portion of the depth map decreases with a large number of views. Thus, a higher percentage of bits, saved from the lateral views, would be dedicated to enhance the quality of the central view.
c) The general ConvNet model works well for all sequences (within and outside the training set), which shows the generality of the proposed method.

TABLE II

COMPARISON OF THE PERFORMANCE, FOR VIEW $1-5$, BETWEEN PROPOSED METHOD AND BENCHMARK. STEPS $15-17$ ARE SKIPPED IN THE PROPOSED METHOD, THUS ONLY GAINS FROM CONVNET ASSISTED QUALITY ENHANCEMENT MODEL ARE OBTAINED. FOR THE PROPOSED METHOD, BOTH GENERAL AND SEQUENCE-SPECIFIC CONVNET MODEL ARE TESTED, WITH THE LATER ONE IN BOLD.

sequence	*benchmark*							*Proposed (Step 1-14)*							$\overline{PSNR}$ *gain (dB)*
	QP assignment					*Total bits*	$\overline{PSNR}$ *(dB)*	*QP assignment*					*Total bits*	$\overline{PSNR}$ *(dB)*	
	v1	*v2*	*v3*	*v4*	*v5*			*v1*	*v2*	*v3*	*v4*	*v5*			
Kendo	51	37	29	37	51	1909	40.95	51	43	28	42	51	1743	41.58	0.63
								51	**45**	**28**	**44**	**51**	**1661**	**41.57**	**0.62**
	48	33	26	33	48	3094	42.60	51	40	25	38	51	2697	43.03	0.43
								51	**40**	**24**	**39**	**50**	**3051**	**43.49**	**0.89**
	46	32	25	32	46	3649	43.13	50	39	24	37	50	3170	43.52	0.39
								51	**39**	**23**	**38**	**50**	**3475**	**43.85**	**0.72**
Undo-dancer	51	38	30	38	51	4912	36.12	51	47	29	46	51	4310	36.89	0.77
								51	**46**	**29**	**46**	**51**	**4347**	**36.92**	**0.80**
	49	34	27	34	49	8520	38.07	51	43	26	42	50	7388	38.88	0.81
								51	**43**	**26**	**42**	**51**	**7370**	**38.90**	**0.83**
	46	32	25	32	46	12441	39.48	51	41	23	40	50	12317	40.89	1.41
								51	**41**	**23**	**40**	**51**	**12299**	**40.92**	**1.44**
Balloons	51	38	30	38	51	1467	40.65	51	44	28	44	51	1361	41.15	0.50
	51	35	28	35	51	1904	41.66	51	40	26	40	51	1831	41.87	0.21
	47	33	26	33	47	2532	42.50	51	38	24	38	51	2457	42.60	0.10
GTfly	51	38	30	38	51	1932	38.93	51	45	29	45	51	1706	39.56	0.63
	47	33	26	33	47	4161	40.65	51	40	25	40	51	3873	41.22	0.57
	46	32	25	32	46	5133	41.10	51	39	24	39	51	4856	41.65	0.55
Average						4305	40.49						3976	41.07	0.58
						5754	**40.06**						**5367**	**40.94**	**0.88**

TABLE III

COMPARISON OF THE PERFORMANCE, FOR VIEW $2-4$, BETWEEN PROPOSED METHOD AND BENCHMARK. STEPS $15-17$ ARE SKIPPED IN THE PROPOSED METHOD, THUS ONLY GAINS FROM CONVNET ASSISTED QUALITY ENHANCEMENT MODEL ARE OBTAINED. FOR THE PROPOSED METHOD, BOTH GENERAL AND SEQUENCE-SPECIFIC CONVNET MODEL ARE TESTED, WITH THE LATER ONE IN BOLD.

sequence	*benchmark*					*Proposed (Step 1-14)*					$\overline{PSNR}$ *gain (dB)*
	QP assignment			*Total bits*	$\overline{PSNR}$ *(dB)*	*QP assignment*			*Total bits*	$\overline{PSNR}$ *(dB)*	
	v2	*v3*	*v4*			*v2*	*v3*	*v4*			
Kendo	41	30	41	1364	40.68	46	29	46	1293	41.36	0.68
						48	**29**	**48**	**1235**	**41.36**	**0.68**
	37	27	37	2139	42.31	43	26	42	2050	42.88	0.57
						45	**26**	**44**	**1968**	**42.88**	**0.57**
	33	24	33	3480	43.88	40	23	39	3256	44.28	0.40
						41	**23**	**40**	**3197**	**44.27**	**0.39**
Undo-dancer	42	31	42	3435	35.69	49	30	49	3351	36.44	0.75
						49	**30**	**48**	**3378**	**36.46**	**0.77**
	37	27	37	7091	38.24	45	26	45	6885	39.08	0.84
						45	**26**	**45**	**6885**	**39.09**	**0.85**
	34	25	34	10773	39.72	43	24	42	10549	40.58	0.86
						43	**24**	**43**	**10488**	**40.58**	**0.86**
Balloons	43	32	43	931	39.93	49	30	49	893	40.87	0.94
	40	29	40	1295	41.40	47	28	47	1127	41.72	0.32
	37	27	37	1734	42.32	44	26	44	1530	42.47	0.15
GTfly	41	31	41	1419	38.72	48	30	48	1310	39.30	0.58
	36	27	36	2968	40.39	43	26	43	2920	40.94	0.55
	35	26	35	3630	40.82	35	26	35	3630	41.03	0.21
Average				3355	40.34				3233	40.91	0.57
				4714	**40.08**				**4525**	**40.77**	**0.69**

TABLE IV
THE PSNR GAIN OF LATERAL VIEWS OBTAINED WITH FINAL PROCESS (STEP $12-14$ IN ALGORITHM 1, WHICH ASSISTS TO FULLY EXPLOIT THE BANDWIDTH) FOR 5 VIEWS SCENARIO OF UNDODANCER.

sequence	*Bandwidth (bits)*	*Step 1-14*							*Step 1-17*							$\overline{PSNR}$ *Gain of lateral views (dB)*
		v1	*v2*	*v3*	*v4*	*v5*	*Total bits*	$\overline{PSNR}$ *of lateral views (dB)*	*v1*	*v2*	*v3*	*v4*	*v5*	*Total bits*	$\overline{PSNR}$ *of lateral views (dB)*	
Undo-dancer	4912	51	47	29	46	51	4310	35.16	50	43	29	44	51	4597	35.67	0.51
		51	**46**	**29**	**46**	**51**	**4347**	**35.32**	**49**	**44**	**29**	**43**	**49**	**4658**	**35.77**	**0.45**
	8520	51	43	26	42	50	7388	36.54	48	40	26	41	49	7775	36.82	0.28
		51	**43**	**26**	**42**	**51**	**7370**	**36.64**	**49**	**40**	**26**	**41**	**48**	**7774**	**36.92**	**0.28**
	12441	51	41	23	40	50	12317	37.37	51	40	23	40	50	12402	37.45	0.08
		51	**41**	**23**	**40**	**51**	**12299**	**37.51**	**50**	**40**	**23**	**40**	**51**	**12402**	**37.58**	**0.07**
Average	8624						8005 *(93%)*	36.36						8258 *(96%)*	36.65	0.29
							8005 *(93%)*	**36.49**						**8278** *(96%)*	**36.76**	**0.27**

3) Gains from the exploitation of the Remaining Available Bits (Steps $15-17$*):* All the steps in the Algorithm 1 are executed in this section. With the steps $15-17$, the remaining bits, i.e. the difference between the allocated bits after running steps $1-14$ and the available bandwidth, are allocated to the lateral views depending on the effectiveness of ConvNet-based quality enhancement.

The obtained gains are shown in Table IV.

In this experiment, $\Re$ is calculated according to Eq. (2). Undodancer is tested as an example with the switching range of 5 views. The changes in the QP values for the lateral views are underlined. It can be found that, 18 out of 24 QPs are decreased, and the remaining are unchanged. Accordingly, the percentage of used bandwidth increases in general from 93% to 96% for both the general ConvNet model and the sequence-specific ConvNet model. The overall PSNR for the lateral views, as well as the overall PSNR gain, are presented. On average, a 0.3 dB gain is reported. The gains of steps $15-17$ become more significant when bandwidth is scarce.

V. CONCLUSION

In this paper, a convolutional neural network assisted multi-view video streaming system is proposed. The system consists of two parts: a ConvNet assisted multiview representation method and a navigation guided bit allocation mechanism. The former representation method removes dependencies between different views to provide an increased flexibility. At the same time, redundancies among views are reduced and exploited with the assistance of the ConvNet model, thus leading to an increase in compression efficiency. As for the proposed bit allocation mechanism, it optimizes the overall quality within the throughput bound, with seamless view switching ability. These two modules can be incorporated in any multiview video streaming system to provide a satisfactory viewing experience within the bandwidth constraint.

For future work, different switching scenarios in the navigation model could be investigated, which can be classified as having high, medium and low dynamism. High movement comes with a higher switching probability than those of the other two categories, which leads to a wider range of views that might be watched. The challenges brought along will be investigated. MVD coding might also be considered in future works, especially when given a high view switching probability.

REFERENCES

[1] I. Sexton and P. Surman, "Stereoscopic and autostereoscopic display systems," *IEEE Signal Processing Magazine*, vol. 16, no. 3, pp. 85–99, May 1999.

[2] Z. Zhang, R. Wang, C. Zhou, Y. Wang, and W. Gao, "A compact stereoscopic video representation for 3d video generation and coding," in *2012 Data Compression Conference*, April 2012, pp. 189–198.

[3] Z. Zhang, C. Zhou, R. Wang, Y. Wang, and W. Gao, "A compact representation for compressing converted stereo videos," *IEEE Transactions on Image Processing*, vol. 23, no. 5, pp. 2343–2355, May 2014.

[4] A. Vetro, T. Wiegand, and G. J. Sullivan, "Overview of the stereo and multiview video coding extensions of the h.264/mpeg-4 avc standard," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 626–642, April 2011.

[5] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *2007 IEEE International Conference on Image Processing*, vol. 1, Sept 2007, pp. I – 201–I – 204.

[6] A. Smolic, K. Mueller, P. Merkle, C. Fehn, P. Kauff, P. Eisert, and T. Wiegand, "3d video and free viewpoint video - technologies, applications and mpeg standards," in *2006 IEEE International Conference on Multimedia and Expo*, July 2006, pp. 2161–2164.

[7] G. J. Sullivan, J. M. Boyce, Y. Chen, J. R. Ohm, C. A. Segall, and A. Vetro, "Standardized extensions of high efficiency video coding (hevc)," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 1001–1016, Dec 2013.

[8] K. Mller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, F. H. Rhee, G. Tech, M. Winken, and T. Wiegand, "3d high-efficiency video coding for multi-view video and depth data," *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 3366–3378, Sept 2013.

[9] Z. Pan, Y. Zhang, and S. Kwong, "Efficient motion and disparity estimation optimization for low complexity multiview video coding," *IEEE Transactions on Broadcasting*, vol. 61, no. 2, pp. 166–176, June 2015.

[10] T. Maugey, I. Daribo, G. Cheung, and P. Frossard, "Navigation domain representation for interactive multiview imaging," *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 3459–3472, Sept 2013.

[11] J. Xiao, M. M. Hannuksela, T. Tillo, M. Gabbouj, C. Zhu, and Y. Zhao, "Scalable bit allocation between texture and depth views for 3-d video streaming over heterogeneous networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 1, pp. 139–152, Jan 2015.

[12] D. Ren, S. H. G. Chan, G. Cheung, and P. Frossard, "Coding structure and replication optimization for interactive multiview video streaming," *IEEE Transactions on Multimedia*, vol. 16, no. 7, pp. 1874–1887, Nov 2014.

[13] T. Maugey and P. Frossard, "Interactive multiview video system with low decoding complexity," in *2011 18th IEEE International Conference on Image Processing*, Sept 2011, pp. 589–592.

[14] M. M. Hannuksela, D. Rusanovskyy, W. Su, L. Chen, R. Li, P. Aflaki, D. Lan, M. Joachimiak, H. Li, and M. Gabbouj, "Multiview-video-plus-depth coding based on the advanced video coding standard," *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 3449–3458, Sept 2013.

[15] E. Kurutepe, M. R. Civanlar, and A. M. Tekalp, "Client-driven selective streaming of multiview video for interactive 3dtv," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 11, pp. 1558–1565, Nov 2007.

[16] Y. Chen, Y.-K. Wang, K. Ugur, M. M. Hannuksela, J. Lainema, and M. Gabbouj, "The emerging mvc standard for 3d video services," *EURASIP Journal on Applied Signal Processing*, vol. 2009, p. 8, 2009.

[17] M. Karczewicz and R. Kurceren, "The sp- and si-frames design for h.264/avc," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 637–644, July 2003.

[18] G. Cheung, A. Ortega, and N. M. Cheung, "Interactive streaming of stored multiview video using redundant frame structures," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 744–761, March 2011.

[19] X. Xiu, G. Cheung, and J. Liang, "Delay-cognizant interactive streaming of multiview video with free viewpoint synthesis," *IEEE Transactions on Multimedia*, vol. 14, no. 4, pp. 1109–1126, Aug 2012.

[20] T. Maugey and P. Frossard, "Interactive multiview video system with low complexity 2d look around at decoder," *IEEE Transactions on Multimedia*, vol. 15, no. 5, pp. 1070–1082, Aug 2013.

[21] J. Chakareski, V. Velisavljevi, and V. Stankovi, "User-action-driven view and rate scalable multiview video coding," *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 3473–3484, Sept 2013.

[22] B. Gu, V. S. Sheng, K. Y. Tay, W. Romano, and S. Li, "Incremental support vector learning for ordinal regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 7, pp. 1403–1416, July 2015.

[23] X. Wen, L. Shao, Y. Xue, and W. Fang, "A rapid learning algorithm for vehicle classification," *Information Sciences*, vol. 295, pp. 395 – 406, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025514010238

[24] C. Wang, J. Zhou, and S. Liu, "Adaptive non-local means filter for image deblocking," *Signal Processing: Image Communication*, vol. 28, no. 5, pp. 522–530, 2013.

[25] Y. Yang, N. P. Galatsanos, and A. K. Katsaggelos, "Projection-based spatially adaptive reconstruction of block-transform compressed images," *IEEE Transactions on Image Processing*, vol. 4, no. 7, pp. 896–908, 1995.

[26] C. Dong, Y. Deng, C. Change Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 576–584.

[27] Z. Wang, D. Liu, S. Chang, Q. Ling, Y. Yang, and T. S. Huang, "D3: Deep dual-domain based fast restoration of jpeg-compressed images," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[28] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 2802–2810.

[29] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. PP, no. 99, pp. 1–1, 2017.

[30] R. Liao, X. Tao, R. Li, Z. Ma, and J. Jia, "Video super-resolution via deep draft-ensemble learning," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[31] Y. Xie, J. Xiao, T. Tillo, Y. Wei, and Y. Zhao, "3d video super-resolution using fully convolutional neural networks," in *2016 IEEE International Conference on Multimedia and Expo (ICME)*, July 2016, pp. 1–6.

[32] I. Sodagar, "The mpeg-dash standard for multimedia streaming over the internet," *IEEE MultiMedia*, vol. 18, no. 4, pp. 62–67, April 2011.

[33] Y. Zhou, Y. Duan, J. Sun, and Z. Guo, "Towards simple and smooth rate adaption for vbr video in dash," in *Visual Communications and Image Processing Conference, 2014 IEEE*, Dec 2014, pp. 9–12.

[34] J. Xiao, M. M. Hannuksela, T. Tillo, and M. Gabbouj, "A paradigm for dynamic adaptive streaming over http for multi-view video," in *Pacific Rim Conference on Multimedia*. Springer, 2015, pp. 410–418.

[35] M. Zhao, X. Gong, J. Liang, J. Guo, W. Wang, X. Que, and S. Cheng, "A cloud-assisted dash-based scalable interactive multiview video streaming framework," in *Picture Coding Symposium (PCS), 2015*, May 2015, pp. 221–226.

[36] T. Su, A. Javadtalab, A. Yassine, and S. Shirmohammadi, "A dash-based 3d multi-view video rate control system," in *Signal Processing and Communication Systems (ICSPCS), 2014 8th International Conference on*, Dec 2014, pp. 1–6.

[37] T. Nunome and H. Tani, "Multi-view video and audio transmission with mpeg-dash and its qoe," in *2015 21st Asia-Pacific Conference on Communications (APCC)*, Oct 2015, pp. 575–579.

[38] C. Zhou, C. W. Lin, and Z. Guo, "mdash: A markov decision-based rate adaptation approach for dynamic http streaming," *IEEE Transactions on Multimedia*, vol. 18, no. 4, pp. 738–751, April 2016.

[39] L. Yu, T. Tillo, and J. Xiao, "Qoe-driven dynamic adaptive video streaming strategy with future information," *IEEE Transactions on Broadcasting*, vol. PP, no. 99, pp. 1–12, 2017.

[40] L. Zhou, "Qoe-driven delay announcement for cloud mobile media," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 1, pp. 84–94, Jan 2017.

[41] ——, "On data-driven delay estimation for media cloud," *IEEE Transactions on Multimedia*, vol. 18, no. 5, pp. 905–915, May 2016.

[42] P. Ndjiki-Nya, M. Koppel, D. Doshkov, H. Lakshman, P. Merkle, K. Muller, and T. Wiegand, "Depth image-based rendering with advanced texture synthesis for 3-d video," *IEEE Transactions on Multimedia*, vol. 13, no. 3, pp. 453–465, June 2011.

[43] B. Li, J. Xu, D. Zhang, and H. Li, "Qp refinement according to lagrange multiplier for high efficiency video coding," in *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, May 2013, pp. 477–480.

[44] F. L. at Nagoya University, "Nagoya university sequences," http://www.fujii.nuee.nagoya-u.ac.jp/multiview-data/.

[45] "HM (HEVC Test Model)16.0," https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.0/.

[46] L. Fang, N. M. Cheung, D. Tian, A. Vetro, H. Sun, and O. C. Au, "An analytical model for synthesis distortion estimation in 3d video," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 185–199, Jan 2014.